Q1:

In this system, T2 (low priority) starts first at time 0 and grabs the lock. At time 1, T1 (high priority) arrives and wants the lock, but it gets blocked because T2 still holds it. At time 2, T3 (medium priority) arrives and preempts T2, since T3 has a higher priority than T2. Now the high-priority thread T1 is stuck waiting, even though it's the most important. T2, which holds the lock, can't finish because T3 is taking up the CPU. That's priority inversion - T1 is delayed by T3, even though T3 is supposed to have lower priority than T1.

Q2: Priority Inheritance Protocol (PIP)

1. T2 runs from time 0 to 5, so it uses all 5 units without interruption.

2. It finishes by time 5.

3. T2 (0-5), T1 (5-9), T3 (9-12). Total = 12 units.

4. Yes, T1 is delayed until time 5, but not by T3.

5. T3 runs last after T2 and T1 are both done.

Q2: Priority Protection Protocol (PPP)

1. T2 again runs from time 0 to 5, uninterrupted.

2. It finishes by time 5.

3. Same schedule: T2 (0-5), T1 (5-9), T3 (9-12). Total = 12 units.

4. T1 is delayed but T3 doesn't block it.

5. T3 is pushed to the end, runs after T2 and T1.

Virtual and Physical Address Calculations

Configuration	VA	PA	VPN	PPN	Offset
32-bit, 4KB, 1GB RAM	32	30	20	18	12
32-bit, 16KB, 2GB RAM	32	31	18	17	14
64-bit, 16KB, 16GB RAM	64	34	50	20	14

Calculation Work:

32-bit OS, 4KB pages, 1 GB RAM

Virtual address: 32 bits (standard for 32-bit OS)

Page size: 4 KB = 2^12 => Offset = 12 bits

RAM = 1 GB = 2^30 => Physical address = 30 bits

VPN = 32 - 12 = 20 bits

PPN = 30 - 12 = 18 bits

32-bit OS, 16KB pages, 2 GB RAM

Virtual address: 32 bits

Page size: 16 KB = 2^14 => Offset = 14 bits

RAM = 2 GB = 2^31 => Physical address = 31 bits

VPN = 32 - 14 = 18 bits

PPN = 31 - 14 = 17 bits

64-bit OS, 16KB pages, 16 GB RAM

Virtual address: 64 bits (standard for 64-bit OS)

Page size: 16 KB = 2^14 => Offset = 14 bits

RAM = 16 GB = 2^34 => Physical address = 34 bits

VPN = 64 - 14 = 50 bits

PPN = 34 - 14 = 20 bits